AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph that begins on page 2, line 20 with the following amended paragraph:

One choice for dealing with diverse devices is to implement applications in the JAVA™ virtual machine environment [[Java]]. The JAVA [[Java]] virtual machine provides a platform independent environment to host applications. However, even though JAVA [[Java]] works well for servlets and applets on desktop computers, JAVA [[Java]] is missing some infrastructure that is needed to support mobile execution of services. In particular, JAVA [[Java]] is lacking in support of transparent remote storage and disconnected operation.

Please replace the paragraph that begins on page 4, line 15 with the following amended paragraph:

FIG. 2 is a functional block diagram that illustrates the interposition of an interposed class for an original class in accordance with one embodiment of the invention. In one embodiment of the invention, standard JAVA [[Java]] application programming interface (API) classes and methods are extended without modifying either the application source code or the [[Java]] JAVA virtual machine and standard API library. For a JAVA~~Java~~-based implementation, JAVA [[Java]] byte-codes are modified at load time and prior to resolution, such that standard JAVA [[Java]] application programming interface (API) class intantiations and method invocations are replaced by instantiations and invocations of extensions of the original class or substitutes of the original methods.

Please replace the paragraph that begins on page 5, line 21 with the following amended paragraph:

In the general case, classes 158 are classes that will be transformed by interposition of the interposed classes 170. For example, in a specific embodiment, classes 158 are selected JAVA [[Java]] system classes that are transformed. The interposed classes 170 are used in lieu of or as extensions of the selected JAVA [[Java]] system classes. It will be appreciated that in one embodiment, the interposed classes are installed on the client as a "middleware" software layer. Application 152 refers to the interposed classes 170 instead of classes 158 (e.g., the JAVA [[Java]] system classes).

Please replace the paragraph that begins on page 5, line 28 with the following amended paragraph:

The following description describes an example implementation of the present invention. The example implementation interposes substitute classes and methods for standard JAVA [[Java]] API classes and methods. The particular classes and methods that are interposed are selected to address various issues relating remote storage, disconnection, and concurrency within a single JAVA [[Java]] virtual machine.

Please replace the paragraph that begins on page 6, line 1 with the following amended paragraph:

Management of the user's data in a mobile environment impacts the JAVA [[Java]] implementation. For example, access to distant resources (JAVA [[Java]] classes, user data, URLs) must be detected and locally cached so that disconnection is a non-fatal event and performance remains acceptable. In order for an end-user's personal data and profiles to be available on the possible devices at the user's disposal, the data must be persistent and securely stored. This requirement implies a third party storage provider to store and retrieve the data. in

the JAVA [[Java]] implementation, an objective is to support legacy services without imposing any software changes to support the remote storage. Many embedded and mobile devices do not facilitate remote storage. Because applications use the standard java.io package to store data, the JAVA [[Java]] implementation is arranged such that when methods are invoked from this package, the files are transparently loaded, refreshed and updated on the remote storage server. Similarly methods such as java.awt.Toolkit.getImage are redirected to the remote storage server.

Please replace the paragraph that begins on page 6, line 14 with the following amended paragraph:

Disconnection issues also impact the JAVA [[Java]] implementation. Continuous internet connectivity can be expensive and interruptions in service can be expected. Some types of services or applications may proceed locally if the user's data and URLs are cached on the client device. The cache content is regularly synchronized and flushed if required. IN the example embodiment, the JAVA [[Java]] implementation

Please replace the paragraph that begins on page 6, line 19 with the following amended paragraph:

Table 1 briefly summarizes the standard JAVA [[Java]] APIs that are modified.

Please replace the paragraph that begins on page 7, line 2 with the following amended paragraph:

The following paragraphs present example JAVA [[Java]] source code in which substitute classes and methods are interposed. The examples are presented for illustration only, and it will be clear from the discussion accompanying FIG. 2 that the application source is not required. The interposition is accomplished instead by modification to the class files.

Please replace the paragraph that begins on page 11, line 29 with the following amended paragraph:

Even though the invention is described in terms of service infrastructure such as JAVA [[Java]], those skilled in the art will appreciate that teachings of the present invention could be adapted to other infrastructures, such as the .NET platform from Microsoft. It will also be appreciated that the invention is applicable to application programs that are not downloadable. In addition to the example embodiments described above, other aspects and embodiments of the present invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and illustrated embodiments be considered as examples only, with a true scope and spirit of the invention being indicated by the following claims.